Gigamon® ThreatINSIGHT

Developing an Intelligence-Driven Threat Hunting Methodology

Joe Slowik, Gigamon Applied Threat Research

Table of Contents

Introduction

Threat hunting emerged as a popular concept for network defense over the past several years. While emphasized in vendor marketing and industry literature, the concept of "threat hunting" remains somewhat slippery in terms of its precise meaning. Furthermore, precise guidance and requirements for building a successful threat hunting program are hard to find.

This paper seeks to address the diffusion of generally shallow descriptions of cyber threat hunting by rigorously defining the concept, then exploring its requirements for success. Through this examination, we will identify an intelligence-driven, hypothesis-based methodology for threat hunting that provides a high-level framework for organizations and defenders to adapt to their specific circumstances.

In addition to interrogating threat hunting and developing a conception of its process, this paper will conclude with an argument on threat hunting's purpose. Rather than simply representing a continuous, manual endeavor, this paper will take the position that threat hunting is a critical initial step in the development of longer-lasting, automated threat detection. While initial hunts may reveal intrusions previously missed, the output of this process will be defined as closing detection gaps to prevent future evasions from taking place.

Defining “Hunting”

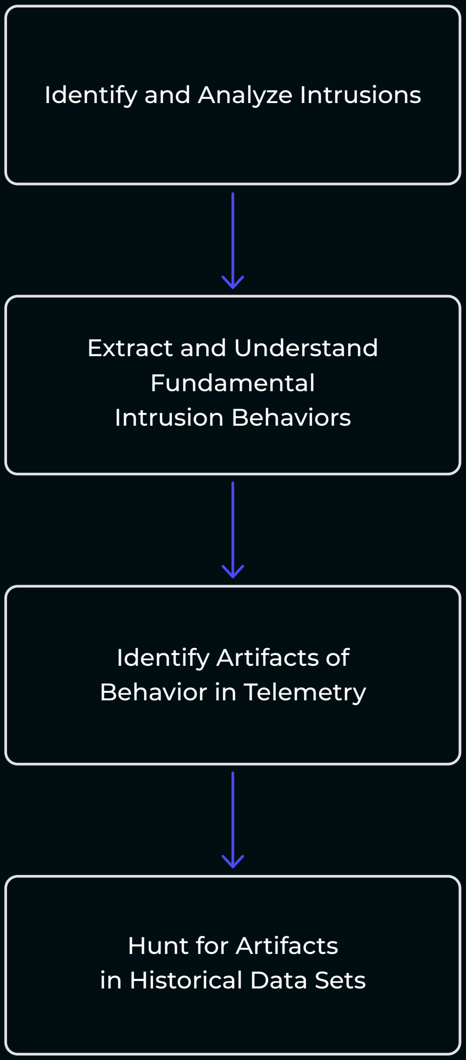
Threat hunting in computer and network defense features many definitions, but all typically coalesce around the same general idea: a human-driven process to identify artifacts associated with a previously undetected intrusion or breach that was not identified by existing security controls.[1] Among other methodologies, threat hunting is often correlated with historical search of indicators of compromise (IOCs) within the defended environment. At times, this may extend to signature-like indicators of attack (IOAs) as well.[2]

These approaches can be effective in unearthing instances of known threat actor activity and should not be immediately discounted. Yet an IOC (and even IOA)-based approach to threat hunting is limited by the very nature of such artifacts: They are historical observations of adversary actions at a given, past point in time.[3]

As with discussions of cyber threat intelligence (CTI) more generally, wherever possible, defenders should work to examine adversary behaviors and methodologies over static, historical indicators related to such activity. Admittedly, not all organizations will achieve this level of security maturity — and may lack the telemetry necessary to even adopt a more robust approach, a consideration that will be explored later. Yet an effective approach to threat hunting should incorporate a reasonably robust understanding not just of specific adversary actions, but how future instances of that activity may look in more general data sources in the future.

Figure 1.

OVERVIEW OF THE HUNTING PROCESS


Identify and Analyze Intrusions
Extract and Understand Fundamental Intrusion Behaviors
Identify Artifacts of Behavior in Telemetry
Hunt for Artifacts in Historical Data Sets

As viewed in the above diagram, an effective threat hunting program begins with analysis of adversary operations and intrusions. Defenders can then identify fundamental behaviors, described using a common methodology, such as the MITRE ATT&CK framework, for purposes of consistency,[4] and look for expected technical observations linked to such behaviors.

Functionally, this process of analysis, enrichment, and understanding is equivalent to methodologies necessary in CTI pivoting and indicator analysis processes.[5] Artifacts of known, historical intrusions are used as initial observations to unearth and understand more fundamental behaviors and tendencies underlying the specific observation. The results of analysis should therefore be more generalized observations of underlying adversary operations, and not specific instances or examples of such behaviors like an IOC.

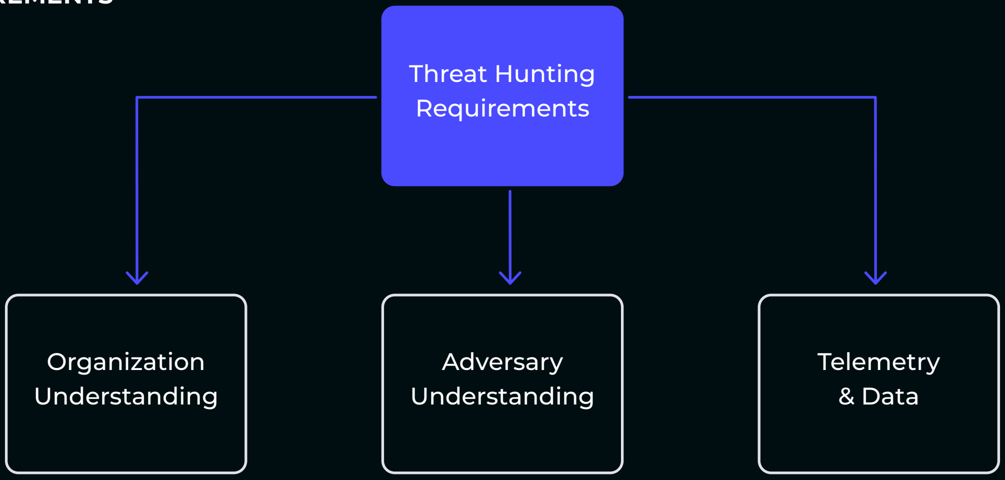
The above provides an initial framework for a more robust definition of threat hunting activity: analysis of available data to identify observations linked to known adversary behaviors or tradecraft to reveal previously undiscovered intrusions. This approach avoids the rigidity of an IOC-based approach and allows greater flexibility in identifying future variations of adversary tools, techniques, and procedures. Yet, although helpful, this definition still leaves much to the imagination as the actual performance of a threat hunt requires several critical prerequisites to ensure success.

Hunting Prerequisites

In the previous section, we identified a reasonably robust, if simple, definition of threat hunting as focused on adversary behaviors and tradecraft. Yet this definition is insufficient to guide practitioners toward a successful threat hunting endeavor. Instead, several prerequisite items are necessary to ensure viable, accurate, sustainable hunting activity. Hunting represents a desirable end state for mature security organizations. Yet requirements to achieve an effective, sustainable hunting posture may put this beyond the reach of many entities. By understanding needed items to build an effective hunting program, security leaders and practitioners can grasp necessary investments and improvements to grow and mature the organization in the desired direction.

Figure 2.

HUNTING REQUIREMENTS



ADVERSARY UNDERSTANDING

The first and most obvious prerequisite to hunting is understanding adversary operations. While seemingly clear, adversary understanding — in the form of CTI — is often misunderstood and misapplied, leading to suboptimal outcomes for hunting purposes. As noted in previous sections, organizations can implement a primitive hunting-like program through ingestion of IOCs and searching for these in available telemetry. This may represent the best that many organizations can hope to achieve within current capabilities but limits observations to specific, known examples of adversary activity. Furthermore, IOC searching absent further understanding may result in queries, or "proto-hunts," for activity of little or no relation to the organization, wasting time and resources on items of little relevance.

For example, a security program can ingest "threat intel" on ransomware operations consisting of previously observed command and control (C2) network infrastructure and ransomware sample hashes. Yet the technical indicators are likely related to a specific campaign or even a very specific victim (especially in the case of sample hash values), making their utility for hunting purposes very limited. An organization can search for these IOCs in their environment, find no evidence of their existence, and then arrive at a false sense of security given lack of highlighted observations.

Instead of this indicator-driven and dependent approach, a robust hunting program needs to identify *how* adversaries of interest operate instead of chasing past examples of such operations. While the latter may occasionally yield results, for a mature security program, the indicator approach will fail to identify the same underlying actions that use or produce new specific technical identifiers.

In the previous ransomware-related example, instead of focusing on specific IOCs related to ransomware samples and C2, organizations should focus on how the intrusion progressed and what general observables relate to operations. Understanding the ransomware affiliate involved and how they leverage certain intrusion techniques, from use of C2 frameworks such as Cobalt Strike or Sliver to lateral movement mechanisms such as credential replay or remote process execution,[6,7,8] becomes critical in fueling a threat hunting approach. By looking for these techniques instead of specific examples of techniques, organizations can regain a degree of flexibility in hunting operations to catch variants of specific operations within their own environments.

In addition to the above technical analysis of operations to understand behaviors of interest, organizations must also understand adversaries in terms of relevance to the organization. While organizations do not have the

ability to pick and choose their adversaries, security programs nonetheless can prioritize certain threats over others based on the entity's organization and business purpose (covered in a following section). From an adversary-focused perspective, security teams cannot (and should not) track all possible threats, but should focus efforts and scarce resources on those most relevant to the organization and potentially most impactful to the organization's continued operation.

Thus, a process of refinement is necessary for security operations to limit ingestion and analysis of behaviors and adversary operations to those most relevant to the organization's risk and threat profile. Establishing intelligence priorities and requirements outlining areas of focus and interest can focus adversary research and understanding to those entities most relevant, ensuring security resources are focused on those entities most impactful or significant to the organization's own security profile. By investing effort in understanding how these relevant adversaries operate, organizations can then establish hunting methodologies focused on threats of most immediate interest and likely exposure.

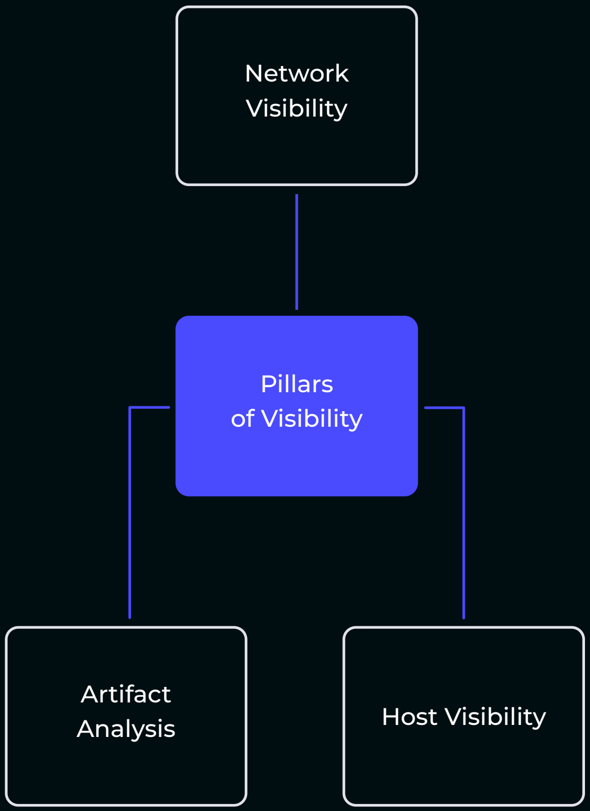
TELEMETRY AND DATA

Understanding the adversary is necessary, but without telemetry to search through, such knowledge becomes useless. Visibility into security events and the related data, along with retention of that data, become critical aspects of threat hunting operations and essential items in structuring hunts. For defenders and prospective hunters, telemetry and data divide into the availability of appropriate sources to query, the ability to effectively query such sources, and the timeliness of such queries to return relevant results.

The simplest and most easily grasped aspect of hunting telemetry is its availability — the presence of logs, data, and other resources against which hunting queries can be executed. Yet not all sources are the same, and understanding the differences among source types and the need for source diversity are critical components of building a successful hunt program.

Figure 3.

THE PILLARS OF VISIBILITY



As shown in the above diagram, we can envision visibility in computer security as consisting of three primary pillars:

- **Network visibility,** such as NetFlow, proxy logs, or firewall logs
- **Host visibility,** such as system logs, endpoint detection and response (EDR) platform data, or Windows event logs
- **Artifact analysis,** such as anti-malware solutions, email filters, and other devices analyzing files or specific payloads

Overemphasis on (or outright limitation to) a single pillar results in a circumscribed, limited perspective of security events of interest. Ideally, searching, detection engineering, and hunting development involve and cross-correlate events among two or more pillars to enrich activity and increase analytical confidence. In this fashion, analysts can gain insight into more complex behaviors involving multiple steps or adversary actions.

For example, typical security alerting or minimal hunting may look for evidence of traffic to a certain IP address provided by a third party as an IOC. A more complex and effective hunt would instead look for signs of the underlying behavior associated with the specific IOC. In this case, a threat hunter may query across all three pillars of visibility to look for network traffic from new or unknown IP ranges resulting in the transfer of an unknown, unsigned binary file resulting in execution from the %TEMP% directory of a given host. While relatively simplistic, this sequence of events represents a significant improvement over single IOC searches in terms of effectiveness, flexibility, and ability to identify variations of adversary operations beyond specific technical examples.

The above, however, remains contingent on the accessibility and timeliness of the data in question. For an effective hunting program to work, underlying information within a given security pillar must be searchable in an effective fashion, ideally in such a way that allows for joins or cross-pivoting between various data sources. Products such as security information and event management (SIEM) systems may enable some of this, but in many cases, effective cross-source combinations may rely on getting data out of initial sources and performing database-like manipulations to join and link events into composite structures.

Furthermore, the accessibility of such data must be timely — in two critical components. First, searches should return in a reasonable timeframe to allow for effective, efficient searches and hunts in near real time. Second, data should be retained for a sufficient duration to enable effective hunting across organizational history. However, these two features may exist in conflict with one another, as fast and efficient search becomes increasingly difficult the longer data is retained, while longer retention imposes greater costs in both storage and indexing requirements.

In this case, organizations must arrive at a sensible balance between features matching the type of threats with which the entity is most concerned. If one is most concerned with complex, long-running state-sponsored espionage activity, longer retention may be necessary to catch and investigate such campaigns, whereas a focus on cybercrime and ransomware operations may require emphasis on quickly implemented searching to catch rapidly unfolding intrusions before they can be monetized. Looking at concrete examples, NOBELIUM campaigns from 2020 involved potentially years of activity for some victims,[9, 10] while ransomware operations may unfold in as little as four hours (although this rapid progression appears anomalous).[11]

BUSINESS VALUE AND IMPACT SCENARIOS

As suggested in the previous two sections, a nascent hunting program must apply some degree of focus and decision-making to determine what adversaries or behavior types are of most interest and how to effectively architect and design telemetry collection and search to capture activity of concern. At this stage, hunting program design becomes introspective in working to understand what relevant cyber impact scenarios exist for the organization, what adversaries align with these scenarios, and what data sources and capabilities are needed to identify such activity.

First, security program leaders and hunting program designers must understand organization operations and requirements to appropriately determine cyber “touch points” that may put these value centers at risk. Identifying things like critical paths for organization operation, critical technologies and services, and key resources enabling continuity of operation can identify points of emphasis for visibility as well as highlighting what adversaries (or adversary types) are most significant to the specific entity.

Second, identification of key resources and capabilities can then enable identification of intrusion scenarios of specific interest and concern. For some organizations, maintaining confidentiality of customer data or proprietary information may be of utmost importance, while for others, continuity of system availability and functionality is most critical. These scenarios and effective threat modeling around these situations will reveal what items are most important for threat hunters to focus on in designing hunting criteria and plans.

Through exploration and understanding of critical internal processes, business values, and their related impact scenarios, threat hunters achieve greater focus and build necessary context for threat modeling. This type of modeling is a necessary intermediate step to ensure CTI-driven adversary research and security engineering-focused telemetry collection and system design align with what the defended organization needs. Absent this step, the prior two sections of understanding adversary behaviors and developing appropriate sources of data will be unmoored from operational realities, and hunting programs will lack appropriate focus to build effective, relevant threat hunting.

Various mechanisms exist to implement the above in practice. One popular formulation is through "Crown Jewels Analysis" modeling, where an organization works to identify critical assets for sustaining and accomplishing the entity's mission.[12] Additional, business-oriented models such as critical path analysis or other operations research methodologies may also enter into play in this endeavor.[13] Irrespective of methodology, a successful threat hunting program will require identification of key assets and resources to drive threat understanding and prioritization.

Hunting in Practice

With prerequisites established and satisfied, organizations can begin implementing hunting programs within their environment. Threat hunting is not simply querying a dataset and evaluating results, however. Effective threat hunting represents a very deliberate sequence of events leveraging the resources highlighted previously to ask testable, measurable questions of available data for signs of certain activity of interest.

The following sections will review hunting as a documented, repeatable process. While the items presented here are relatively abstract, this is a necessity, as specific implementation of hunting methodologies is contingent on the threats, operations, and visibility of a specific organization. Nonetheless, by adhering to the following guidelines, a mature security program can implement a documented, sustainable hunting program supporting everyday security operations.

STRUCTURING HYPOTHESES

Hunting is not an arbitrary practice when executed effectively, but instead requires deliberate formulation of hunting hypotheses.[14] Our three hunting prerequisites inform this process by providing different considerations to feed into our hunting process:

1. **What adversary or threat actor behaviors are of interest?**

2. **How can these behaviors impact my organization?**

3. **What data sources exist that yield artifacts related to adversary operations?**

The answers to these questions should form the building blocks of a specific, testable statement: our preliminary (and at this time, provisional) hypothesis. An analyst should arrive at a statement related to the threat activity of interest that allows for timely evaluation, or testing, within available data.

For this exercise, we structure items in the same fashion as statistical hypothesis testing: making an assumption based upon the three items above, collecting evidence to evaluate the assumption, and then determining whether to accept or reject the assumption based on findings.[15] For the purposes of information security threat hunting, we (typically) do not need to proceed to rigorous statistical analysis of results for evaluation purposes — although organizations with large, diverse datasets (such as security product providers with access to telemetry from multiple installations) may pursue such evaluations with interesting results.

To use an example, we can look to a fairly common security problem — such as business email compromise (BEC) — and formulate a hypothesis based on what we know and understand of this activity and its relationship to our organization. In the case of BEC, we can hypothesize that malicious messages will inject into or spoof existing communication streams with an attempt to elicit a financial transaction. This relates to adversary actions (spoofing email communications) and why this matters to the organization (a potential financial transaction) — but work is required to build out the testing side of matters, which links to our available data sources and structuring queries around them.

TRANSLATED HYPOTHESES TO TESTABLE QUERIES

Once we arrive at a reasonable hypothesis, it must be translated into specific propositions to test its accuracy or veracity. The nature and design of any test queries will necessarily be determined by the telemetry sources available to a threat hunter. Ideally, hypothesis development should take note of existing capabilities and limitations to ensure that one does not develop untestable hypotheses (e.g., a host-oriented assumption in an environment with limited host-based telemetry). In most cases, a given adversary behavior or action will reflect across network, host, and artifact data sources, allowing for some (if incomplete) options to test even in visibility-poor environments.

Queries should be designed to further emphasize the testable, specific nature of the given hypothesis while distilling the hypothesis down into particular use case scenarios. These can either focus on a specific data source or blend observations across sources for a more composite, behavioral view. Data accessibility in this case becomes critical — being able to interrogate relevant datasets and, ideally, cross-link between data types to arrive at more context-aware views is a powerful strategy to observe complex behaviors.

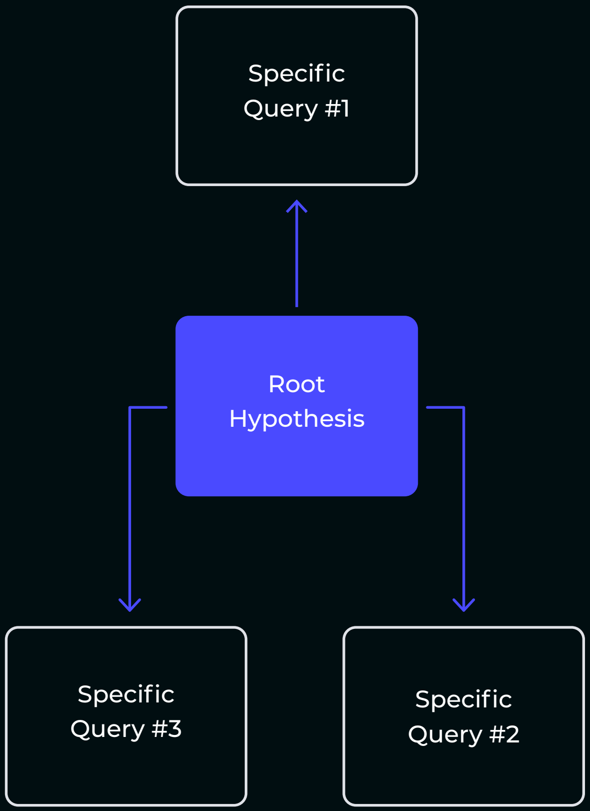
In execution, there should ultimately be a one-to-many relationship between a hypothesis and the various queries designed to test the validity and accuracy of that hypothesis, as illustrated above. While certainly limited to the tools and telemetry available to the organization performing threat hunting, a skilled threat hunter should nonetheless work to exercise as many possible observations of behaviors linked to the hypothesis to ensure thorough and reasonably complete testing.

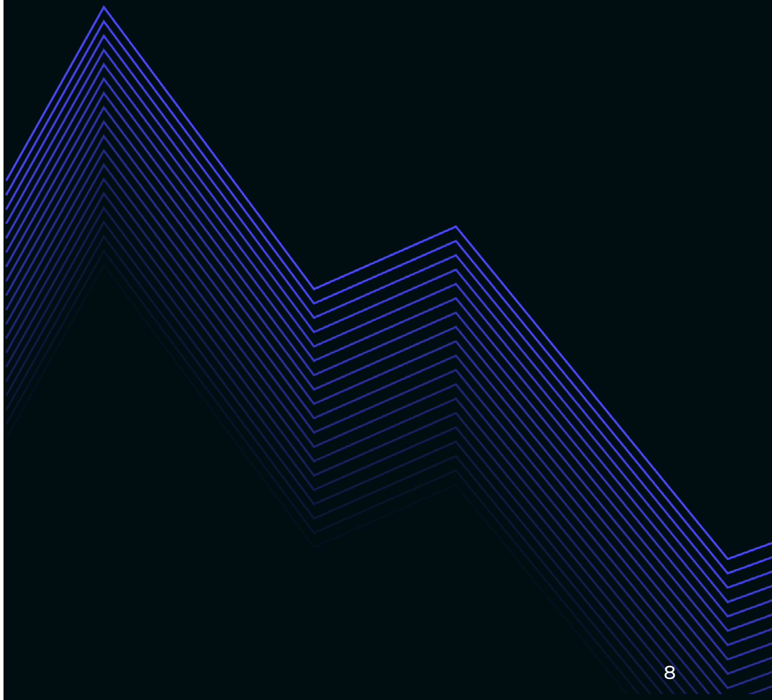
Looking back to the BEC example discussed previously, our general hypothesis of injecting messages into conversations or spoofing existing threads for financial gain can yield multiple, specific technical observations or queries:

- **Identify instances where sender addresses are spoofed** or differ from displayed addresses in conversation threads
- **Identify message threads where messages initially use legitimate DKIM values,** followed by invalid or absent DKIM values
- **Identify messages with sender addresses or internal link domains matching malicious**

Figure 4.

HYPOTHESIS-TO-QUERY RELATIONSHIP





characteristics (e.g., recently registered, unusual hosting) or that appear to spoof legitimate items

Each of the above items requires some work to implement and varying degrees of visibility into email traffic, metadata, and content. But through assembling multiple potential queries to test the hypothesis, a threat hunter can devise various perspectives on the problem to test (and potentially validate) the hypothesis.

EVALUATING RESULTS

Following query design and implementation, threat hunters must evaluate results. Simply put, evaluation translates into determining whether the queries unearthed evidence or instances of the activity inspiring the hypothesis. Of note, some queries may prove to be successful translations of the underlying hypothesis, while others may fail or simply reside beyond implementation for technical reasons for the given organization.

Overall, the process of evaluation takes several steps:

1. **Determine completeness and results of a given query to validate the query's function**

2. **Determine relationship between query results and the hypothesis generating the query**

3. **Based on multiple queries, evaluate the validity of the foundational hypothesis**

A threat hunter will therefore look at several "tiers" of evaluation in a rigorous hunting exercise to validate individual technical queries and, ultimately, the originating hypothesis. This endeavor can lead to multiple potential results, from identifying potential telemetry issues or gaps through revision of individual testing queries to ensure better and more accurate performance. Ultimately, each evaluative step must relate back to understanding of and testing the hypothesis driving the entire exercise.

Of importance in evaluation, threat hunters must understand their results and take a behavior-centric approach to their output. Most significantly, colloquial use of the term "false positive" in security frequently associates such instances with benign examples of a given event or behavior.[16]

For example, a threat hunt for lateral movement techniques may turn up instances of an engineer performing maintenance tasks resulting in rapid authentication to multiple hosts. While the activity in question is benign, it is not a "false positive" if the originating query was designed to look for this type of behavior. Instead, the context around the behavior renders this observation as benign, but in different circumstances the same behaviors could be present and associated with abuse or malicious implementation.

A threat hunter must maintain awareness of these distinctions, especially when dealing with abuse of otherwise legitimate system functionality, so that queries (and potentially even a hypothesis) are not mistakenly rejected due to "high false positives." Instead, threat hunters will need to seek ways to enrich observations with greater detail and contextuality to differentiate between benign examples of the targeted behavior and malicious instances.

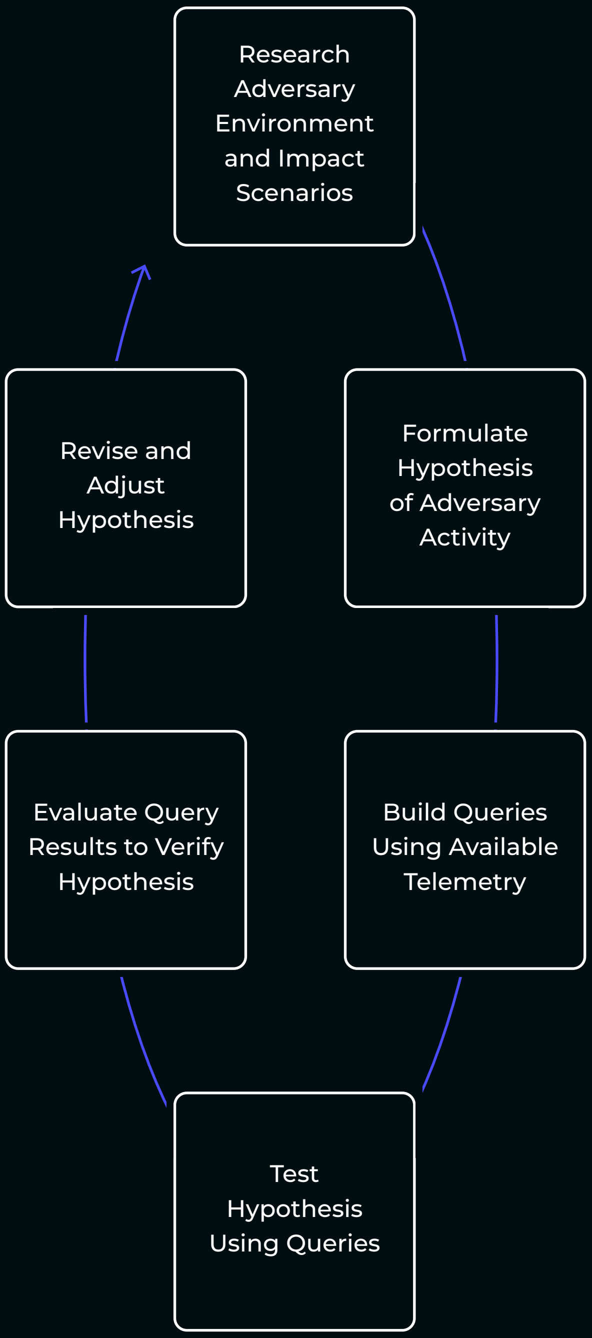
ITERATION AND REFINEMENT

Finally, threat hunting represents a continuous, cyclical process and not a linear endeavor. Each threat hunt, whether successful or otherwise, can and should inform subsequent actions, either by refining an understanding of a given adversary or behavior, or through lessons learned concerning available telemetry and query possibilities. By incorporating lessons learned from completed threat hunts into subsequent actions, threat hunters can develop and implement a process of continued learning and refinement, as displayed in the following diagram.

For threat hunting to move beyond an isolated, "one-off" action and toward a sustainable process, implementing hunting processes as an iterative cycle is not just desirable but necessary. Such actions should also be accompanied by documentation, knowledge transfer, and communication of lessons learned to avoid repeating mistakes and to take complete advantage of past actions. Although at times cumbersome, taking advantage of ticketing systems, wikis, and other resources for documenting and tracking activity is a critical function in enabling a sustainable, long-lived hunting operation free of unnecessary rework and duplication.

Figure 5.

THE ITERATIVE HUNTING PROCESS


Research Adversary Environment and Impact Scenarios
Formulate Hypothesis of Adversary Activity
Build Queries Using Available Telemetry
Test Hypothesis Using Queries
Evaluate Query Results to Verify Hypothesis
Revise and Adjust Hypothesis

When designed and executed as a continuous process, threat hunting transitions from something done in “spare” or otherwise unallocated time to a mature function within the security team. In addition to ensuring threat hunting learns from its own past successes and failures, an iterative and continuous approach to hunting also fuels incorporation of hunting outputs into wider security programs.

Transition Hunts to Detections

At the start of this paper, we examined hunting as a mechanism to overcome gaps in threat detection by providing a methodology to look for items missed by standard security monitoring practices. While this represents a natural starting point for hunting — to fill in holes in the standard security program — this relationship is not our ideal end. Instead, a robust and mature threat hunting program can serve as an input to (or even an inspiration for) the security program and work to close holes in everyday detection and monitoring.

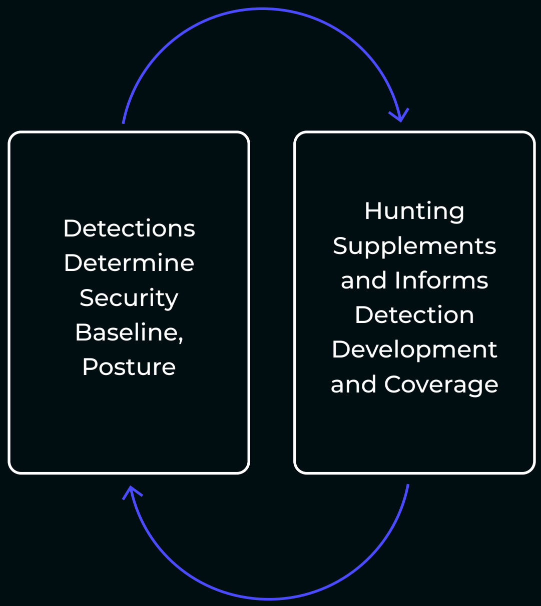
Everyday security operations are typically founded upon alerts or detections running in the background. When activity of interest takes place, the alert fires prompting a responder, typically a SOC analyst, to triage and disposition the item. Detection engineering fuels this process by developing and deploying alerts based upon an understanding of threats and the defended environment[17] — similar background criteria for threat hunting. Given this shared background, well-designed security programs can look to detection engineering and threat hunting as related disciplines, with opportunities for each to inform the other.

From the perspective of most mature security organizations, threat hunting and detection engineering can represent mutually reinforcing functions. While detection engineering focuses on long-term, system-enabled identification of malicious activity, threat hunting seeks to identify unique or atypical items missed or otherwise not seen in implemented security monitoring. Yet when threat hunting identifies a high-fidelity, useful query through hypothesis testing, detection engineers can and should look to implement such items into the organization’s corpus of detections and alarms.

Through this implementation of threat hunting as a support to more traditional detection engineering, threat hunting can deliver long-lasting benefits to

Figure 7.

THE RELATIONSHIP BETWEEN DETECTIONS AND HUNTING


Detections Determine Security Baseline, Posture
Hunting Supplements and Informs Detection Development and Coverage

security operations. While identifying intrusions otherwise missed in a threat hunt is certainly valuable, making meaningful additions to the set of automated security detections of the organization represents a long-running "win" for the defended entity.

By translating successful, meaningful queries from threat hunting investigations to threat detections, hunters can encode knowledge and expertise from hunts into daily security workflows. As part of the hypothesis testing process, threat hunters can and should look for opportunities to translate hunting queries into long-running threat detections. If hunting queries have sufficient fidelity and accuracy, their utility for security operations in identifying future malicious behavior should be clear and easy to grasp.

Through a continuous dialog between threat hunting and detection engineering, organizations can combine an open-ended search for malicious behaviors with long-running detection development and deployment to achieve a more complete security posture. While requiring additional work, from both threat hunters and detection engineers, a unified process of detections and hunting feeding each other provides for a robust, continuously adaptive security monitoring stance.

Conclusion

Threat hunting remains a popular concept in information security, but one rarely explored to any significant degree of detail or refinement. In this paper, we presented a methodology for creating and implementing a hunting program at a sufficient level of abstraction for most mature security programs to be able to build something useful from the provided items. In taking the lessons communicated in this article and committing them to the security program, organizations can both close detection gaps in their environment while also boosting detection development, creating a more complete and resilient organization.

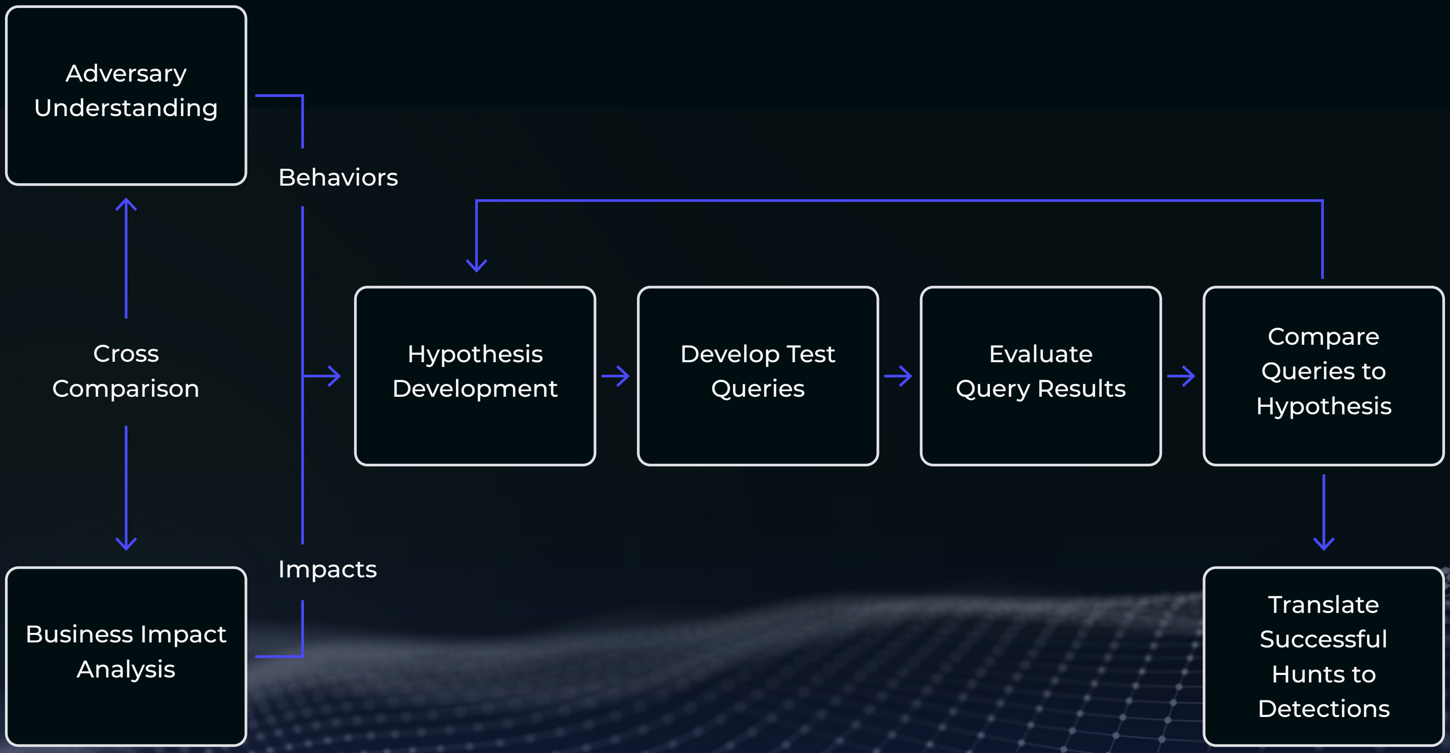
While the above represents an ideal, we also recognize that not all organizations possess the necessary prerequisites to engage in this activity. The defined requirements for threat hunting — adversary understanding, business impact analysis, and proper telemetry — may be well beyond the means and budget of many organizations that nonetheless face significant security concerns. For these entities, a “proto-hunting” posture of indicator searching and reacting to events may represent the best possible set of circumstances at this time.

Although the programmatic advice in this paper may not be universally applicable given the foundations upon which a successful hunting program must be built, all organizations can extract meaningful lessons from this exploration. Principally, security programs require a combination of internal and external understanding along with a recognition of capabilities and existing visibility to successfully operate. Should any of these be absent, an organization is not only unable to implement a sustainable threat hunting practice, but even more basic security operations may become difficult or impossible to effectively implement.

Thus, threat hunting represents an ideal toward which organizations should strive. While not easily reached, the requirements for creating and implementing a successful threat hunting program equate to critical capabilities that, when implemented, enable and improve more basic security functions. By identifying the needs of a hunting program, even immature security organizations can identify priorities for investment and growth to build more robust and effective programs. Once the necessary security foundation is built, as determined by the three general prerequisites outlined previously, organizations can begin implementing and executing more advanced security practices to counter adaptive and impactful cyber threats.

Figure 8.

FINALIZED HUNTING PROCESS DIAGRAM



About Gigamon

Gigamon offers a deep observability pipeline that harnesses actionable network-level intelligence to amplify the power of observability tools. This powerful combination enables IT organizations to assure security and compliance governance, speed root-cause analysis of performance bottlenecks, and lower operational overhead associated with managing hybrid and multi-cloud IT infrastructures. The result: modern enterprises realize the full transformational promise of the cloud. Gigamon serves more than 4,000 customers worldwide, including over 80 percent of Fortune 100 enterprises, nine of the 10 largest mobile network providers, and hundreds of governments and educational organizations worldwide.

To learn more, please visit gigamon.com.

Resources

[1] S. Taschler, “What Is Cyber Threat Hunting?,” March 15, 2022. CrowdStrike. https://www.crowdstrike.com/cybersecurity-101/threat-hunting/.

[2] Kaspersky, “Kaspersky,” July 6, 2021. https://support.kaspersky.com/KATA/3.7/en-US/194907.htm. Accessed April 28, 2022.

[3] J. Slowik, “Indicators and Network Defense,” May 16, 2018. https://pylos.co/2018/05/16/indicators-and-network-defense/. Accessed April 28, 2022.

[4] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington and C. B. Thomas, “MITRE ATT&CK: Design and Philosophy,” March 2020. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf. Accessed July 28, 2022.

[5] J. Slowik, “Forumulating a Robust Pivoting Methodology,” 2021, DomainTools. https://pylos.co/wp-content/uploads/2021/02/pivoting.pdf. Accessed July 22, 2022.

[6] A. Rahman, “Defining Cobalt Strike Components So You Can BEA-CONfident in Your Analysis,” October 12, 2021. Mandiant. https://www.mandiant.com/resources/defining-cobalt-strike-components. Accessed July 28, 2022.

[7] BishopFox, “Sliver: Cross-Platform General Purpose Implant Framework Written in Golang,” https://bishopfox.com/tools/sliver. Accessed July 28, 2022.

[8] M. Lazic, “Investigating Lateral Movement - WMI and Scheduled Tasks,” February 3, 2022. Gigamon. https://blog.gigamon.com/2022/02/03/investigating-lateral-movement-wmi-and-scheduled-tasks/. Accessed July 28, 2022.

[9] J. Lambert, “The Hunt for NOBELIUM, the Most Sophisticated Nation-State Attack in History,” November 10, 2021. Microsoft. https://www.microsoft.com/security/blog/2021/11/10/the-hunt-for-nobelium-the-most-sophisticated-nation-state-attack-in-history/. Accessed July 28, 2022.

[10] J. Slowik, “Continuous Eruption: Further Analysis of the SolarWinds Supply Chain Incident,” December 18, 2020. DomainTools. https://www.domaintools.com/resources/blog/continuous-eruption-further-analysis-of-the-solarwinds-supply-incident. Accessed July 28, 2022.

[11] The DFIR Report, “Quantum Ransomware,” April 25, 2022. https://thedfirreport.com/2022/04/25/quantum-ransomware/. Accessed July 28, 2022.

[12] MITRE, “Crown Jewels Analysis.” https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/crown-jewels-analysis. Accessed July 28, 2022.

[13] F. K. Levy, G. L. Thompson, and J. D. Wiest, “The ABCs of the Critical Path Method,” November 1963. Harvard Business Review. https://hbr.org/1963/09/the-abcs-of-the-critical-path-method. Accessed July 28, 2022.

[14] R. M. Lee and D. Bianco, “Generating Hypotheses for Successful Threat Hunting,” 2021. Sans Institute. https://sansorg.egnyte.com/dl/qyBaLJHovj. Accessed July 28, 2022.

[15] Penn State University, “Statistics Online - Hypothesis Testing.” https://online.stat.psu.edu/statprogram/reviews/statistical-concepts/hypothesis-testing. Accessed July 25, 2022.

[16] J. Slowik, “Revisiting the Idea of the “False Positive”,” August 5, 2022. https://blog.gigamon.com/2022/08/05/revisiting-the-idea-of-the-false-positive/. Accessed TBD August 2022.

[17] J. Day, “So, You Want to Be a Detection Engineer?” February 24, 2020. Gigamon. https://blog.gigamon.com/2020/02/24/so-you-want-to-be-a-detection-engineer/. Accessed July 30, 2022.

Gigamon®

Worldwide Headquarters
3300 Olcott Street, Santa Clara, CA 95054 USA
+1 (408) 831-4000 | gigamon.com